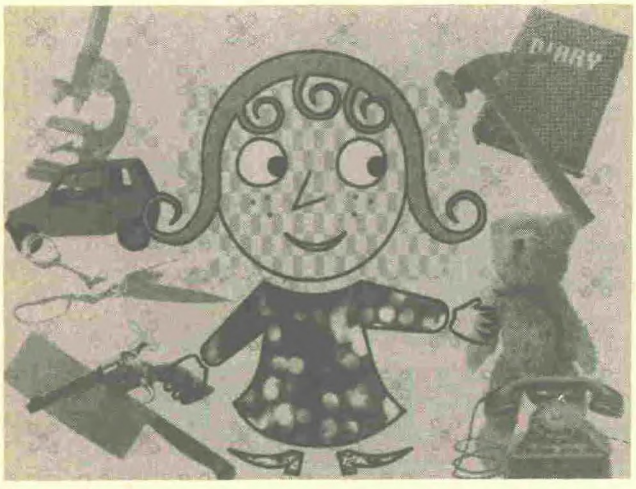
ณรงค์ จันทโพธิ์

SOFT MAIL

สร้างสรรค์การใช้งานเม้าส์ด้วย OOP : โปรแกรมวาดรูป



สร้างโปรแกรมวาดรูป
สำหรับช่วยสร้างรูปภาพ
เพื่อเอาไว้ใช้งานต่อไป

ในครั้งแรกผมตั้งใจว่าตอนนี้จะพูดถึงปุ่มกดแบบมีรูปภาพ แต่ผมเกรงว่าท่านผู้อ่านจะมีปัญหาในการเตรียมรูปภาพเพื่อใช้ในการสร้าง ดังนั้นในบทความตอนนี้ผมจะขอนำออบเจ็กต์ที่ได้ออกแบบไว้แล้วจากตอนที่ผ่านมา สร้างโปรแกรมที่ช่วยสร้างรูปภาพเอาไว้ใช้ในบทความตอนหน้าก็แล้วกันครับ

ในโปรแกรมนี้ผมแบ่งออกเป็น 3 ส่วน คือ ส่วนของเมนู ส่วนของจานสี และส่วนของพื้นที่วาดรูป โดยแต่ละส่วนนี้ผมมองเป็นออบเจ็กต์แต่ละตัว ดังนั้นจะเห็นว่าต้องมีการสร้างออบเจ็กต์ขึ้นมาใช้งานเฉพาะในโปรแกรมนี้ นั่นคือส่วนของจานสีและส่วนของพื้นที่วาดรูป โดยผมขอกล่าวถึงลักษณะการทำงานในออบเจ็กต์ทั้งสองตัวนี้ก่อนครับ

จานสี

1. เนื่องจากการเขียนโปรแกรมในกราฟิกโหมดของภาษา C ที่ความละเอียด 640x480 pixel นี้ จะสามารถแสดงสีได้เพียง 16 สี ดังนั้นในส่วนของจานสีนี้ จะมีสีให้เลือกเพียง 16 สี

2. ในการเลือกสีใช้งานนั้นสามารถเลือกได้ครั้งละ 2 สี โดยการคลิกเม้าส์ปุ่มขวา 1 สี และคลิกเม้าส์ปุ่มซ้ายอีก 1 สี

3. จะมีส่วนแสดงสีที่ถูกเลือกทั้ง 2 สีในแต่ละครั้งของการเลือกสีใหม่

พื้นที่วาดรูป

1. พื้นที่วาดรูปนี้จะมี 2 ส่วน คือ ส่วนที่ใช้ในการวาดรูป และส่วนแสดงภาพขนาดเท่าจริง ส่วนที่เราจะทำงานได้นั้นจะเป็นส่วนที่ใช้ในการวาดรูปเท่านั้น ในส่วนนี้ผมได้กำหนดขนาดไว้เป็น 32 x 32 pixel

2. ในการวาดนั้นจะสามารถใช้เม้าส์ทั้งปุ่มซ้ายและปุ่มขวา โดยสีที่วาดนั้นจะเป็นไปตามสีที่เลือกจากจานสี

จากลักษณะการทำงานข้างต้น ผมจะขอนำไปสู่แนวทางในการออกแบบออบเจ็กต์ทั้ง 2 ตัว เริ่มต้นมาดูแนวทางการออกแบบจานสีครับ ส่วนแรกที่จะต้องทำการออกแบบคือหน้าตาของจานสี ซึ่งแล้วแต่ความพอใจของแต่ละท่าน สำหรับผมสร้างให้แต่ละสีมีขนาด 15 x 15 pixel โดยแบ่งเป็น 2 แถว ในการวาดนั้นก็ใช้ Loop ทำการวาดแต่ละสีดังนี้

```
for(i=0;i<2;i++)
for(j=0;j<8;j++){
setfillstyle(1,c++);
bar(SCol+(j*CBS)+j,SRow+(i*CBS)+i,
SCol+(j*CBS)+CBS+j,
SRow+(i*CBS)+CBS+i);DeepBar
(SCol+(j*CBS)+j,
SRow+(i*CBS)+i,CBS,CBS);
 }
```

ซึ่งจะมีส่วนแสดงสีพื้นหน้าและสีพื้นหลังที่ได้เลือกไว้ด้วย การทำงานในส่วนการวาดจานสีนี้ จะอยู่ในส่วนของ Method ColorPalette::Draw() ครับ

ในการตอบสนองต่อการคลิกเม้าส์นั้น (การเลือกสี) อย่างที่ได้กล่าวไปแล้วว่าสามารถใช้ได้ทั้งปุ่มขวาและปุ่มซ้ายในการเลือก ดังโค้ดต่อไปนี้ เป็นการตอบสนองต่อการคลิกเม้าส์ปุ่มซ้ายซึ่งเป็นการเลือกสีพื้นหน้า

```
if((MouEvent==L_DOWN)&&(MouInBox(MouX,MouY,SCol,
SRow,ECol,ERow))){
HideMouse();
ForeColor = getpixel(MouX,MouY); //อ่านค่าสีพื้นหน้า
UpdateColor();
ShowMouse();
…
}
```

และสำหรับการเลือกสีพื้นหลังจะเป็นดังโค้ดต่อไปนี้ครับ

```
if((MouEvent==R_DOWN)&&(MouInBox(MouX,MouY,SCol,
SRow,ECol,ERow))){
HideMouse();
BkColor = getpixel(MouX,MouY);//อ่านสีค่าพื้นหลัง
UpdateColor();
ShowMouse();
…
}
```

จากลิสติงข้างต้นนี้ซึ่งเป็นเพียงบางส่วนใน Method ColorPalette:: Pressed() จะใช้ฟังก์ชัน getpixel() ในการอ่านค่าสี ณ ตำแหน่งที่เม้าส์ถูกกด ดังนั้นในการเลือกสีนี้จะต้องกดปุ่มเม้าส์ในพื้นที่สีที่ต้องการเท่านั้น ถ้าหากกดในบริเวณขอบของสีนั้น (ซึ่งผมออกแบบเป็นกรอบสามมิติสีเทา) จะทำให้ได้สีที่เพี้ยนไปจากที่ต้องการได้ หากท่านผู้อ่านไม่ต้องการให้เกิดเหตุการณ์ดังกล่าวอาจจะหาวิธีการอ่านค่าสีที่ทำการเลือกใหม่ เช่น คำนวณตำแหน่งของเม้าส์ที่กดออกมาเป็นหมายเลขของค่าสี หลังจากการเลือกสีสิ้นสุดลงจะมีการแสดงสีที่ได้ทำการเลือกใหม่โดยทำการเรียก Method ColorPalette::UpdateColor() มาทำงาน

ก่อนอื่นก็ต้องกล่าวถึงส่วนแสดงหน้าตาก่อน ก็เช่นเดิมครับผู้ที่ทำหน้าที่ในการวาดหน้าตาของส่วนพื้นที่วาดรูป ก็ต้องเป็น Method DrawBoard::Draw() ที่ผมได้ออกแบบไว้นี้จะมีลักษณะเป็นตารางขนาด 24 x 24 ช่อง โดยที่สีพื้นจะเป็นไปตามสีพื้นที่เลือกไว้ โดยที่เริ่มต้นผมกำหนดไว้เป็นสีขาว

สำหรับการตอบสนองการกดเม้าส์นั้นหากกดปุ่มซ้ายจะเป็นการวาดด้วยสีพื้นหน้าและถ้ากดปุ่มขวาจะเป็นการวาดด้วยสีพื้นหลัง การทำงานในส่วนนี้จะเป็นหน้าที่ของ Method DrawBoard::Pressed() ดังนี้

```
   int DrawBoard::Pressed(){
while((MouEvent=L_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow))){
   HideMouse();
   GetCoordinate(&MouX,&MouY);
   UpdateCell(MouX,MouY,ForeColor);
   ShowMouse();
   GetMouInfo(&MouEvent,&MouX,&MouY);
   }
while((MouEvent=R_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow))){
   HideMouse();
   GetCoordinate(&MouX,&MouY);
   UpdateCell(MouX,MouY,BkColor);
   ShowMouse();
   GetMouInfo(&MouEvent,&MouX,&MouY);
   }
   setcolor(ForeColor);
   return 0;
}
```

จากลิสติงข้างต้นนี้ เมื่อพบว่ามีการกดเม้าส์ในส่วนของพื้นที่วาดรูปก็จะทำการคำนวณหาตำแหน่งของ Cell ที่จะทำการระบายโดยการเรียก Method DrawBoard::GetCoordinate() มาทำงาน เมื่อได้ตำแหน่งของ Cell แล้วก็จะทำการระบายสีของ Cell นั้นโดยการเรียก Method DrawBoard::UpdateCell() มาทำงาน

สำหรับโครงสร้างของรูปที่ทำการวาดนั้นจะเก็บไว้ที่ตัวแปร Bitmap ซึ่งประกาศไว้เป็น Array ขนาด 24 x 24 แต่ละช่องของ Array จะใช้เก็บค่าสี 1 ค่า ที่ได้จากการวาดบนพื้นที่วาดรูป โค้ดที่สมบูรณ์ของจานสีและพื้นที่วาดรูปอยู่ในลิสติงที่ 1 ดังนี้

ลิสติงที่ 1
PALETTE.CPP

```
#include<graphics.h>
#include<conio.h>
#include"stdfon.h"
#include"mouobj.h"
#include"objects.h"
#include"palette.h"
#define CBS 19
extern WORD MouEvent;
extern int MouX,MouY;
extern int ForeColor,BkColor;
extern int Bitmap[24][24];
int First=1;
//*****************************************************************************
void DeepBar(int x,int y,int w,int h){
  setcolor(WHITE);
  rectangle(x-1,y-1,x+w+1,y+h+1);
  setcolor(DARKGRAY);
  line(x,y,x+w,y);
  line(x,y,x,y+h);
  setcolor(LIGHTGRAY);
  line(x+w,y+1,x+w,y+h);
  line(x+1,y+h,x+w,y+h);
  setcolor(BLACK);
  line(x+1,y+1,x+w-1,y+1);
  line(x+1,y+1,x+1,y+h-1);
}
```

→

ลิสติงที่ 1 (ต่อ)

```
//*****************************************************************************
ColorPalette::ColorPalette(int x,int y){
  SCol  = x+2*CBS;
  SRow  = y;
  ECol  = SCol + 8*CBS;
  ERow  = y + 2*CBS;
  Style = VISIBLE;
  Draw();
}
//*****************************************************************************
void ColorPalette::Draw(){
int i,j,c=0;
  setfillstyle(1,WHITE);
  bar(SCol-(CBS*2)-2,SRow,SCol-2,SRow+(CBS*2));
  DeepBar(SCol-(CBS*2)-2,SRow,CBS*2,CBS*2+1);
  for(i=0;i<2;i++)
    for(j=0;j<8;j++){
       setfillstyle(1,c++);
       bar(SCol+(j*CBS)+j,SRow+(i*CBS)+i,SCol+(j*CBS)+CBS+j,
          SRow+(i*CBS)+CBS+i);
       DeepBar(SCol+(j*CBS)+j,SRow+(i*CBS)+i,CBS,CBS);
       }
   UpdateColor();
}
//*****************************************************************************
void ColorPalette::UpdateColor(){
  setfillstyle(1,LIGHTGRAY);
  bar(SCol-CBS-6,SRow+CBS-3,SCol-8,SRow+(CBS*2)-6);
  setfillstyle(1,BkColor);
  bar(SCol-CBS-4,SRow+CBS-1,SCol-11,SRow+(CBS*2)-9);
  setfillstyle(1,LIGHTGRAY);
  bar(SCol-CBS-13,SRow+CBS-11,SCol-15,SRow+(CBS*2)-14);
  setcolor(DARKGRAY);
  line(SCol-8,SRow+CBS-3,SCol-8,SRow+(CBS*2)-6);
  line(SCol-CBS-7,SRow+(CBS*2)-6,SCol-8,SRow+(CBS*2)-6);

  line(SCol-15,SRow+CBS-11,SCol-15,SRow+(CBS*2)-14);
  line(SCol-CBS-14,SRow+(CBS*2)-14,SCol-15,SRow+(CBS*2)-14);
  setfillstyle(1,ForeColor);
  bar(SCol-CBS-11,SRow+CBS-9,SCol-18,SRow+(CBS*2)-17);
}
//*****************************************************************************
int ColorPalette::Pressed(){
  if((MouEvent==L_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow))){
    HideMouse();
    ForeColor = getpixel(MouX,MouY);
    UpdateColor();
    ShowMouse();
    while((MouEvent==L_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow)))
       GetMouInfo(&MouEvent,&MouX,&MouY);
    return 0;
    }
  if((MouEvent==R_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow))){
    HideMouse();
    BkColor = getpixel(MouX,MouY);
    UpdateColor();
    ShowMouse();
    while((MouEvent==R_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow)))
       GetMouInfo(&MouEvent,&MouX,&MouY);
    return 0;
    }
  return 0;
}
//*****************************************************************************
DrawBoard::DrawBoard(int x,int y,int w,int h){
  Style = VISIBLE;
  SCol = x;
  SRow = y;
  ECol = x + w;
  ERow = y + h;
//   ECol  = SCol + 24*CBS;
//   ERow  = y + 24*CBS;
  Draw();
  First = 0;
}
```

→

ลิสติงที่ 1 (ต่อ)

```
//*****************************************************************************
void DrawBoard::Draw(){
int i,j;
  HideMouse();
  setfillstyle(1,BkColor);
  bar(SCol,SRow,ECol,ERow);
  bar(ECol+30,SRow+20,ECol+30+24,SRow+20+24);
  DeepBar(SCol-2,SRow-2,ECol-SCol+3,ERow-SRow+3);
  DeepBar(ECol+30-2,SRow+20-2,26,26);
  OutThaiXY(ECol+30,SRow,"ขนาดจริง",BLACK);
  OutThaiXY(ECol+30,SRow+60,"ชื่อตัวแปร: ไม่มี",BLACK);
  setcolor(BLACK);
  for(i=0;i<=24;i++){
    line(SCol,SRow+i*10,ECol,SRow+i*10);
    line(SCol+i*10,SRow,SCol+i*10,ERow);
    }
//  if(!First)
  for(i=0;i<24;i++)
    for(j=0;j<24;j++){
      if(First)
        Bitmap[i][j] = BkColor;
      UpdateCell(i,j,Bitmap[i][j]);

      }
  ShowMouse();
}
//*****************************************************************************
void DrawBoard::GetCordinate(int *x,int *y){
  (*x)--;
  (*y)--;
  *x = (*x-SCol)/10;
  *y = (*y-SRow)/10;
}
//*****************************************************************************
void DrawBoard::UpdateCell(int x,int y,int color){
  setfillstyle(1,color);
  bar(SCol+x*10+1,SRow+y*10+1,SCol+x*10+1+8,SRow+y*10+1+8);
  putpixel(ECol+x+30,SRow+y+20,color);
  Bitmap[x][y] = color;
}
//*****************************************************************************
int DrawBoard::Pressed(){
  while((MouEvent==L_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow))){
    HideMouse();
    GetCordinate(&MouX,&MouY);
    UpdateCell(MouX,MouY,ForeColor);
    ShowMouse();
    GetMouInfo(&MouEvent,&MouX,&MouY);
    }
  while((MouEvent==R_DOWN)&&(MouInBox(MouX,MouY,SCol,SRow,ECol,ERow))){
    HideMouse();
    GetCordinate(&MouX,&MouY);
    UpdateCell(MouX,MouY,BkColor);
    ShowMouse();
    GetMouInfo(&MouEvent,&MouX,&MouY);
    }
  setcolor(ForeColor);
  return 0;
}
//*****************************************************************************
```

จากลิสติงที่ 1 นี้ ส่วนของจานสีเป็นออบเจ็กต์ ColorPalette และส่วนของพื้นที่วาดรูปนั้นเป็นออบเจ็กต์ DrawBoard รายละเอียดการทำงานของแต่ละ Method ผมได้กล่าวไปแล้วข้างต้น ยังคงเหลือเพียง Constructor ของแต่ละออบเจ็กต์ที่ยังไม่พูดถึงให้พิจารณาในลิสติงที่ 2 ดังนี้

ลิสติงที่ 2 PALETTE.H

```
#include"objects.h"
class ColorPalette :public AppObj{
public:
  ColorPalette(int,int);
private:
  void Draw();
  void UpdateColor();
  int  Pressed();
};
```

ลิสติงที่ 2 (ต่อ)

```
//************************************************
class DrawBoard :public AppObj{
public:
  DrawBoard(int,int,int,int);
private:
  void UpdateCell(int,int,int);
  void GetCordinate(int *,int *);
  void Draw();
  int  Pressed();
};
//************************************************
```

ColorPalette(int x,int y)

เป็น Constructor ที่ใช้ประกาศสร้างจานสีโดยมี Argument ดังนี้

-int x เป็นตำแหน่งของจานสีทางแกน x

-int y เป็นตำแหน่งของจานสีทางแกน y

DrawBoard(int x,int y)

เป็น Constructor ที่ใช้ประกาศสร้างพื้นที่วาดรูปโดยมี Argument ดังนี้

int x เป็นตำแหน่งของพื้นที่วาดรูปทางแกน x

int y เป็นตำแหน่งของพื้นที่วาดรูปทางแกน y

รูปที่ได้ทำการออกแบบแล้วจะสามารถจัดเก็บลงบนแฟ้มข้อมูล
ได้ 2 รูปแบบ คือ เป็นแบบ Binary โดยที่จะสามารถอ่านขึ้นมาเพื่อแก้ไข
ได้ในภายหลัง และแบบ Text แบบนี้จะได้หลังจากสร้างเป็นที่พอใจแล้ว
จากนั้นสั่งให้โปรแกรมสร้างโค้ดออกมาเป็นรูปแบบของภาษา C รูปแบบ
ของโค้ดที่ได้มีลักษณะของการประกาศตัวแปรของรูปที่ได้ออกแบบไว้
ดังนี้

```
int BitmapName[ ]={24,24,   //ขนาดของรูป 24 x 24 pixel
.
.       //ข้อมูลสีของแต่ละ Pixel จำนวน 24  x 24 ค่า
.
};
```

หลังจากทราบรายละเอียดของออบเจ็กต์กันแล้วเราก็จะมาดูราย
ละเอียดของตัวโปรแกรมหลักดังในลิสติงที่ 3 ดังนี้

ลิสติงที่ 3
DRAW.CPP

```
#include<graphics.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include"mouobj.h"
#include"stdfon.h"
#include"objects.h"
#include"objinter.h"
#include"menu.h"
#include"palette.h"
#define ID_QUIT          10
#define ID_YES           100
#define ID_NO            101
#define ID_OK            100
#define ID_CANCEL  101
#define ID_OPENF   102
#define ID_SAVEF         103
#define ID_GENCODE 104
#define ID_NEW           105
int   ForeColor=BLACK,BkColor=WHITE; //Default draw color
BYTE  FName[50],BitmapName[30];
WORD  MouEvent;
int   MouX,MouY;
GUIApp TestApp;
AppObj *hMenu,*hDrawBrd;
WORD  AppMsg;
int   Quit = 0;
int   Bitmap[24][24];
//*****************************************************************************
int QueryToQuit(){
GUIApp QuitDlg;
WORD QuitMsg;
  QuitDlg.AddObj(new Window("คำเตือน!",150,80,300,150,VISIBLE|
                 CAPTION,WHITE,LIGHTRED,LIGHTGRAY));
  QuitDlg.AddObj(new TextButton(210,190,60,25,"ใช่",VISIBLE|FOCUSED,13,ID_YES));
  QuitDlg.AddObj(new TextButton(330,190,60,25,"ไม่ใช่",VISIBLE,27,ID_NO));
  HideMouse();
  OutThaiXY(190,140,"คุณต้องการออกจากโปรแกรมใช่หรือไม่?",GREEN);
  ShowMouse();
  while(1){       //Wait for CLOSEWIN Message
    GetKbdMouse(&MouEvent,&MouX,&MouY);
    QuitMsg = QuitDlg.GetMessage(MouEvent,MouX,MouY);
    if(QuitMsg==ID_YES)
return 1;
    if(QuitMsg==ID_NO)
return 0;
    }
}
```

ลิสติงที่ 3 (ต่อ)

```
49:
50:  //*************************************************************************
51:  void ShowFName(char *FName){
52:    setfillstyle(1,LIGHTGRAY);
53:    HideMouse();
54:    bar(300,36,500,52);
55:    OutThaiXY(308,36,FName,YELLOW);
56:    ShowMouse();
57:  }
58:  //*************************************************************************
59:  void ShowBitmapName(char *Name){
60:    setfillstyle(1,LIGHTGRAY);
61:    HideMouse();
62:    bar(390,135,600,155);
63:    OutThaiXY(398,140,Name,YELLOW);
64:    ShowMouse();
65:  }
66:  //*************************************************************************
67:  int GetFName(char *title,char *FName){
68:  GUIApp GetF;
69:  WORD Msg;
70:    GetF.AddObj(new Window(title,130,80,350,200,VISIBLE|CAPTION,WHITE,BLUE,
71:                LIGHTGRAY));
72:    GetF.AddObj(new TextButton(200,238,60,25,"ตกลง",VISIBLE|FOCUSED,13,ID_OK));
73:    GetF.AddObj(new TextButton(355,238,60,25,"ยกเลิก",VISIBLE,27,ID_CANCEL));
74:    GetF.AddObj(new LineEdit(150,150,"ชื่อแฟ้ม:",BLUE,30,30,VISIBLE|FOCUSED,1));
75:    while(Msg!=CLOSEWIN){        //Wait for CLOSEWIN Message
76:      GetKbdMouse(&MouEvent,&MouX,&MouY);
77:      Msg = GetF.GetMessage(MouEvent,MouX,MouY);
78:      if(Msg==ID_OK){
79:  strcpy(FName,GetF.GetLnEditStr(1));
80:  return 1;
81:  }
82:       if(Msg==ID_CANCEL)
83:  return 0;
84:       }
85:  }
86:  //*************************************************************************
87:  int GetBitmapName(char *Name){
88:  GUIApp GetBN;
89:  WORD Msg;
90:    GetBN.AddObj(new Window("ตั้งชื่อตัวแปร",130,80,350,200,VISIBLE|CAPTION,
91:                 WHITE,BLUE,LIGHTGRAY));
92:    GetBN.AddObj(new TextButton(200,238,60,25,"ตกลง",VISIBLE|FOCUSED,13,ID_OK));
93:    GetBN.AddObj(new TextButton(355,238,60,25,"ยกเลิก",VISIBLE,27,ID_CANCEL));
94:    GetBN.AddObj(new LineEdit(150,150,"ชื่อตัวแปร:",BLUE,30,30,VISIBLE|FOCUSED,1));
95:    while(1){        //Wait for CLOSEWIN Message
96:      GetKbdMouse(&MouEvent,&MouX,&MouY);
97:      Msg = GetBN.GetMessage(MouEvent,MouX,MouY);
98:      if(Msg==ID_OK){
99:  strcpy(Name,GetBN.GetLnEditStr(1));
100: return 1;
101: }
102:      if(Msg==ID_CANCEL)
103: return 0;
104:      }
105: }
106: //*************************************************************************
107: void Save(char *FName){
108: int i,j;
109: FILE *f;
110:   if((f=fopen(FName,"wb"))==NULL)
111:     return;
112:   putc(24,f);       //Bitmap width
113:   putc(24,f);       //Bitmap height
114:   for(i=0;i<24;i++)
115:     for(j=0;j<24;j++){
116:     putc(Bitmap[i][j],f);
117:     }
118:   fclose(f);
119: }
120: //*************************************************************************
121: void Load(char *FName){
122: int i,j;
123: FILE *f;
```

→

ลิสติงที่ 3 (ต่อ)

```
124:      if((f=fopen(FName,"rb"))==NULL)
125:        return;
126:      getc(f);          //Bitmap width
127:      getc(f);          //Bitmap height
128:      for(i=0;i<24;i++)
129:        for(j=0;j<24;j++){
130:        Bitmap[i][j] = getc(f);
131:        }
132:      fclose(f);
133:    }
134:    //*************************************************************************
135:    void GenCode(char *FName,char *Name){
136:    FILE *f;
137:    int row,col;
138:    char str[3];
139:      if((f=fopen(FName,"wt"))==NULL)
140:        return;
141:      fprintf(f,"char %s[]={24,24,\n",Name);
142:      for(row=0;row<24;row++){
143:        for(col=0;col<24;col++){
144:    itoa(Bitmap[col][row],str,10);
145:    fprintf(f,"%#2s",str);
146:    if(((row*24)+col)<((24*24)-1))
147:      putc(',',f);
148:    }
149:        putc('\n',f);
150:        }
151:      fprintf(f,"};");
152:      fclose(f);
153:    }
154:    //*************************************************************************
155:    void New(){
156:    int i,j;
157:      for(i=0;i<24;i++)
158:        for(j=0;j<24;j++)
159:        Bitmap[i][j] = BkColor;
160:      hDrawBrd->Draw();
161:    }
162:    //*************************************************************************
163:    void DoMessage(WORD *Msg){
164:      switch(*Msg){
165:        case ID_QUIT  : case CLOSEWIN :
166:              Quit = QueryToQuit();
167:              break;
168:        case ID_GENCODE:if(GetBitmapName(BitmapName)&&
169:                 GetFName("จัดเก็บแฟ้ม",FName)){
170:                 ShowBitmapName(BitmapName);
171:                 GenCode(FName,BitmapName);
172:                 }
173:              break;
174:        case ID_SAVEF : if(GetFName("จัดเก็บแฟ้ม",FName)){
175:                 Save(FName);
176:                 ShowFName(FName);
177:                 }
178:              break;
179:        case ID_OPENF : if(GetFName("อ่านแฟ้มข้อมูล",FName)){
180:                 Load(FName);
181:                 ShowBitmapName("");
182:                 hDrawBrd->Draw();
183:                 ShowFName(FName);
184:                 }
185:              break;
186:        case ID_NEW   : New();
187:              break;
188:        }
189:    }
190:    //*************************************************************************
191:    main(){
192:      int Gd=DETECT,Gm;
193:      initgraph(&Gd,&Gm,"c:\\lang\\bc31\\bgi");
194:      InitMouse();
195:      setrgbpalette(EGA_LIGHTGRAY,48,48,48);
196:      setrgbpalette(EGA_DARKGRAY,35,35,35);
197:      TestApp.AddObj(new Window("OOP Programming 7 : โปรแกรมวาดรูป",0,0,639,479,
198:                  VISIBLE|SYS_CLOSE,WHITE,BLUE,LIGHTGRAY));
```

ลิสติงที่ 3 (ต่อ)

```
199:      //Create menu
200:        hMenu = new PopUpMenu("แฟ้มข้อมูล",VISIBLE,1,
201:        new MenuItem("เปิดแฟ้ม...",VISIBLE,ID_OPENF,
202:        new MenuItem("จัดเก็บ",VISIBLE,ID_SAVEF,
203:        new SEPARATOR(
204:        new MenuItem("จบโปรแกรม",VISIBLE,ID_QUIT,NULL)))),
205:      new PopUpMenu("แก้ไข",VISIBLE,2,
206:        new MenuItem("สร้างโค้ด",VISIBLE,ID_GENCODE,
207:        new SEPARATOR(
208:        new MenuItem("สร้างใหม่",VISIBLE,ID_NEW,NULL))),
209:        NULL));
210:      SetUpMenu(hMenu);
211:      hDrawBrd = new DrawBoard(60,80,240,240);
212:      TestApp.AddObj(hDrawBrd);
213:      TestApp.AddObj(new ColorPalette(60,350));
214:      ShowMouse();
215:      while(!Quit){
216:        GetKbdMouse(&MouEvent,&MouX,&MouY);
217:        AppMsg = TestApp.GetMessage(MouEvent,MouX,MouY);
218:        DoMessage(&AppMsg);
219:        if(AppMsg!=ID_QUIT){
220:   AppMsg = GetMenuMessage(hMenu);
221:   DoMessage(&AppMsg);
222:   }
223:        }
224:      DestroyMenu(hMenu);
225:      HideMouse();
226:      closegraph();
227:      return 0;
228:    }
```

บรรทัดที่ 11-19 เป็นการประกาศค่าคงที่ที่ใช้เป็น Message ของแต่ละออบเจ็กต์

บรรทัดที่ 20-28 เป็นการประกาศตัวแปรที่ใช้ในโปรแกรม ซึ่งมีตัวแปรที่ควรกล่าวถึงเป็นพิเศษอยู่ 5 ตัวดังนี้

```
int    ForeColor=BLACK,BkColor=WHITE;
```

เป็นตัวแปรที่ใช้เก็บค่าสีพื้นหน้าและสีพื้นหลังที่ได้จากการใช้เม้าส์เลือกจากจานสี โดยกำหนดค่าเริ่มต้นที่สีพื้นหน้าเป็นสีขาว และสีพื้นหลังเป็นสีดำ

```
BYTE   FName[50],BitmapName[30];
```

Fname เป็นชื่อของแฟ้มข้อมูลที่กำลังแก้ไขในขณะนั้นและ BitmapName จะเป็นชื่อของรูปภาพที่ได้วาดนั้น ตัวแปร BitmapName นี้จะใช้เมื่อมีการสั่งให้ทำการสร้างโค้ดออกมา ในการสร้างโค้ดนั้น จะนำชื่อรูปภาพนี้มาใช้เป็นชื่อตัวแปรในแฟ้มที่ได้จากการสร้างโค้ด

```
int    Bitmap[24][24];
```

เป็นตัวแปรที่ใช้เก็บข้อมูลแต่ละจุดสีของรูปภาพที่ทำการวาด

บรรทัดที่ 30-46 เป็นฟังก์ชันทำหน้าที่ในการถามย้ำถึงความแน่ใจในการเลือกคำสั่งออกจากโปรกรม จะเป็นกรอบข้อความประกอบด้วยปุ่มกด 2 ปุ่ม หากกดปุ่ม "ใช่" จะ return Message ID_YES ออกมา แต่ถ้ากดปุ่ม "ไม่ใช่" จะ return Message ID_NO ออกมาแทน

บรรทัดที่ 51-56 เป็นฟังก์ชันที่ทำหน้าที่ในการแสดงชื่อแฟ้มที่กำลังแก้ไขในขณะนั้น

บรรทัดที่ 59-64 เป็นฟังก์ชันที่ทำหน้าที่ในการแสดงชื่อตัวรูปภาพที่กำลังแก้ไขในขณะนั้น ซึ่งจะถูกเรียกทำงานเมื่อมีการสั่งให้โปรแกรมทำการสร้างโค้ด

บรรทัดที่ 67-85 เป็นฟังก์ชันที่ใช้ในการรับชื่อแฟ้มข้อมูล ในส่วนนี้ผมได้ออกแบบในรูปแบบการทำงานง่าย ๆ ผู้ใช้จะต้องป้อนชื่อแฟ้มเองทั้งหมด

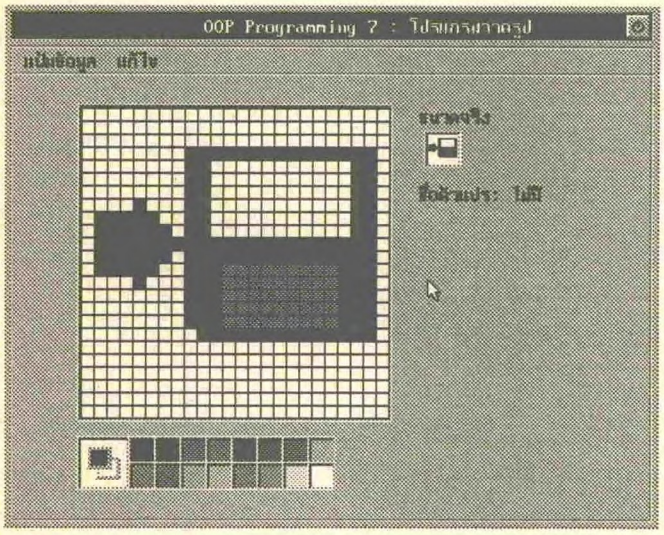
บรรทัดที่ 87-105 เป็นฟังก์ชันที่ใช้ในการรับชื่อของรูปภาพ

บรรทัดที่ 107-118 เป็นฟังก์ชันที่ทำหน้าที่เขียนข้อมูลรูปภาพที่ได้จากการออกแบบจากตัวแปร Bitmap เก็บลงบนแฟ้มข้อมูลในรูปแบบ Binary file ซึ่งจะสามารถอ่านขึ้นมาแก้ไขในภายหลังได้

บรรทัดที่ 121-132 เป็นฟังก์ชันที่ทำหน้าที่อ่านข้อมูลรูปภาพจากแฟ้มข้อมูลในรูปแบบ Binary file ขึ้นมาแก้ไขโดยจะเก็บลงบนตัวแปร Bitmap

บรรทัดที่ 135-152 เป็นฟังก์ชันที่ทำหน้าที่แปรข้อมูลรูปภาพที่ได้จากการออกแบบจากตัวแปร Bitmap ออกมาให้อยู่ในรูปของภาษาซีบนแฟ้มข้อมูลในรูปแบบ Text file ซึ่งจะไม่สามารถอ่านขึ้นมาแก้ไขในภายหลังได้

บรรทัดที่ 155-160 เป็นฟังก์ชันที่ทำหน้าที่ลบส่วนของพื้นที่วาดรูปใหม่เมื่อเลือกคำสั่ง "สร้างใหม่" จากเมนู ในการลบนี้จะใช้สีพื้นหลังที่ได้ถูกเลือกไว้ครั้งหลังสุดเป็นสีพื้น ในส่วนนี้จะเห็นว่ามีการเรียก Method hDrawBrd->Draw(); เพื่อทำการวาดขึ้นมาใหม่นั่นเองที่เป็นเหตุผลที่ต้องทำการสร้างออบเจ็กต์ DrawBaord โดยมี Pointer hDrawBrd เป็นตัวชี้ (ดูในบรรทัดที่ 211)



รูปที่ 1 แสดงหน้าจอการทำงานของโปรแกรม

บรรทัดที่ 163-189 เป็นฟังชันที่ทำหน้าที่ทำงานตาม Message ที่ได้รับจากโปรแกรมโดยจะมีการเรียกฟังก์ชันต่าง ๆ ดังที่ได้กล่าวไปแล้วขึ้นมาทำงาน

บรรทัดที่ 191-228 เป็นตัวโปรแกรมหลักซึ่งพอจะอธิบายแต่ละบรรทัดได้ดังนี้ครับ

บรรทัดที่ 192-196 เป็นการกำหนดเปิด Graphic Mode และกำหนดโทนสีของ LIGHTGRAY และ DARKGRAY ใหม่

บรรทัดที่ 197-198 เป็นการสร้างกรอบหน้าต่างของโปรแกรมหลัก

บรรทัดที่ 200-210 เป็นการกำหนดสร้างเมนูต่าง ๆ

บรรทัดที่ 211 เป็นการสร้างออบเจ็กต์ DrawBoard ขึ้นมาโดยมี Pointer ชี้ เนื่องจากจะมีการเรียกออบเจ็กต์ DrawBoard ขึ้นมาทำงานในภายหลังด้วย หากไม่มี Pointer ชี้ไว้จะทำให้ไม่สามารถเรียกขึ้นมาทำงานได้เลยนอกจากต้องทำการค้นหาใน LinkedList ของออบเจ็กต์ซึ่งจะเป็นการยุ่งยาก ตัวอย่างในการเรียกทำงานดังเช่นในบรรทัดที่ 156 และ 178 ครับ

บรรทัดที่ 212-213 เป็นการเพิ่มออบเจ็กต์ DrawBoard และ ColorPalette เข้าไปใน LinkedList

บรรทัดที่ 215-223 เป็น Message Loop ของโปรแกรม

บรรทัดที่ 224-227 เป็นส่วนของการปิดโปรแกรมโดยจะทำลายเมนูเสียก่อนแล้วจึงจะปิดเม้าส์และปิด Graphic

ปรับปรุงโค้ดเก่า

สำหรับโปรเจ็กต์นี้ไม่มีการปรับปรุงโค้ดเก่ามีแต่การเพิ่มแฟ้มข้อมูล 3 แฟ้ม ดังนี้ครับ

PALETTE.CPP จากลิสติงที่ 1

PALETTE.H จากลิสติงที่ 2

DRAW.CPP จากลิสติงที่ 3

ดังนั้นในโปรเจ็กต์นี้จะมีซอร์สโค้ดอยู่ 11 แฟ้มดังนี้ครับ

1. DRAW.CPP
2. STDFON.CPP
3. MOUOBJ.CPP
4. OBJINTER.CPP
5. BUTTON.CPP
6. CHECKBOX.CPP
7. RADIOBTN.CPP
8. WINDOW.CPP
9. LNEDIT.CPP
10. MENU.CPP
11. PALETTE.CPP

ส่งท้าย

ผมเขียนโปรแกรมตัวอย่างนี้ขึ้นมาเพื่อใช้เป็นตัวอย่างในการใช้งานออบเจ็กต์ต่าง ๆ จากตอนที่ผ่านมาเป็นหลัก ดังนั้นการทำงานของโปรแกรมในบางส่วนอาจจะยังไม่ครบถ้วนจากที่ควรจะเป็นเท่าใดนัก หากท่านผู้อ่านมีความเข้าใจในตัวโปรแกรมและการใช้งานออบเจ็กต์ต่าง ๆ ผมก็คิดว่าไม่ยากเกินไปที่จะเพิ่มส่วนการทำงานอื่น ๆ ให้โปรแกรมทำงานได้สมบูรณ์ยิ่งขึ้น

สำหรับซอฟต์เมล์ก็คงเรียกใช้บริการได้เช่นเดิมครับซึ่งจะประกอบไปด้วยแฟ้มข้อมูลต่าง ๆ ดังนี้

1.OBJECTS.H
2.OBJINTER.H
3.MOUOBJ.H
4.STDFON.H
5.MENU.H
6.PALETTE.H
7.THAIENG1.TXT
8.STDFON.CPP
9.MOUOBJ.CPP
10.RADIOBTN.CPP
11.CHECKBOX.CPP
12.BUTTON.CPP
13.WINDOW.CPP
14.LNEDIT.CPP
15.OBJINTER.CPP
16.MENU.CPP
17.PALETTE.CPP
18.DRAW.CPP

สำหรับท่านผู้อ่านท่านใดมีปัญหาหรือคำแนะนำประการใดสามารถติดต่อผมได้ทาง

Email : ru_boy@poboxes.com
http://www.geocities.com/CollegePark/Union/2029/